PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of: Frank S. Caccavale	Technology Center: 2100
Serial No.: 10/722,146 Confirm: 7027	Group Art Unit: 2195
Filed: 11/25/2003	Examiner: Truong, Camquy
For: Method and Apparatus for Load Balancing of Distributed Processing Units Based on Performance Metrics	Atty. Dkt. No.: 10830.0106.NP

APPEAL BRIEF TO THE BOARD OF PATENT APPEALS AND INTERFERENCES

Commissioner for Patents
PO Box 1450
Alexandria, Virginia 22313-1450

Sir:

The brief is in support of the Notice of Appeal filed on July 10, 2009 appealing from the final Official Action of April 10, 2009. Please charge any deficiency in any required fee to EMC Corporation Deposit Account No. 05-0889.

I. REAL PARTY IN INTEREST

The real party in interest is EMC Corporation, by virtue of an assignment recorded at Reel 014750 Frame 0702.

II. RELATED APPEALS AND INTERFERENCES

There are no related appeals or interferences.

III. STATUS OF THE CLAIMS

Claims 1-36 have been presented for examination.

Claims 1-36 have been finally rejected, and are being appealed.

IV. STATUS OF AMENDMENTS

No amendment was filed after the final Official Action of April 10, 2009.

V. SUMMARY OF CLAIMED SUBJECT MATTER

The invention of appellant's claim 1 is a method in a data processing network (appellant's FIG. 14) including distributed processing units (24, 25, and 26 in FIG. 14; appellant's specification, page 35, lines 1-7). (Specification, page 3, lines 4-5.) The method includes obtaining a respective utilization value of each distributed processing unit (step 162 in FIG. 15; specification, page 35, lines 17-18), applying a mapping function (159 in FIG. 14; specification, page 35, lines 8-11; page 36, lines 1-5; page 37 lines 10-20) to the respective utilization value of said each distributed processing unit to obtain a respective weight for said each distributed processing unit (step 163 in FIG. 15; specification, page 36, lines 1-3), and using the respective weights for the distributed processing units for distributing work requests to the distributed processing units so that the respective weight for said each distributed processing unit specifies a respective frequency at which the work requests are distributed to said each distributed processing unit (step 164 in FIG. 15; specification, page 36 lines 5-7; page 38 lines 1-3). (Specification, page 3, lines 5-12.)

For example, as shown in appellant's FIG. 14, as reproduced below, each NT file server 24, 25, 26 executes a respective program 155, 156, 157 for sensing the percentage utilization $\alpha[i]$ of the NT file server. (Specification, page 35, lines 5-7.) A weighted round-robin load balancing program 158 in the data mover 30 periodically collects the utilizations $\alpha[i]$ from the NT file servers, and accesses a mapping table 159 for converting the utilizations $\alpha[i]$ to a respective set of weights W[i]. (Specification, page 35, lines 8-11.) The weighted round-robin load balancing

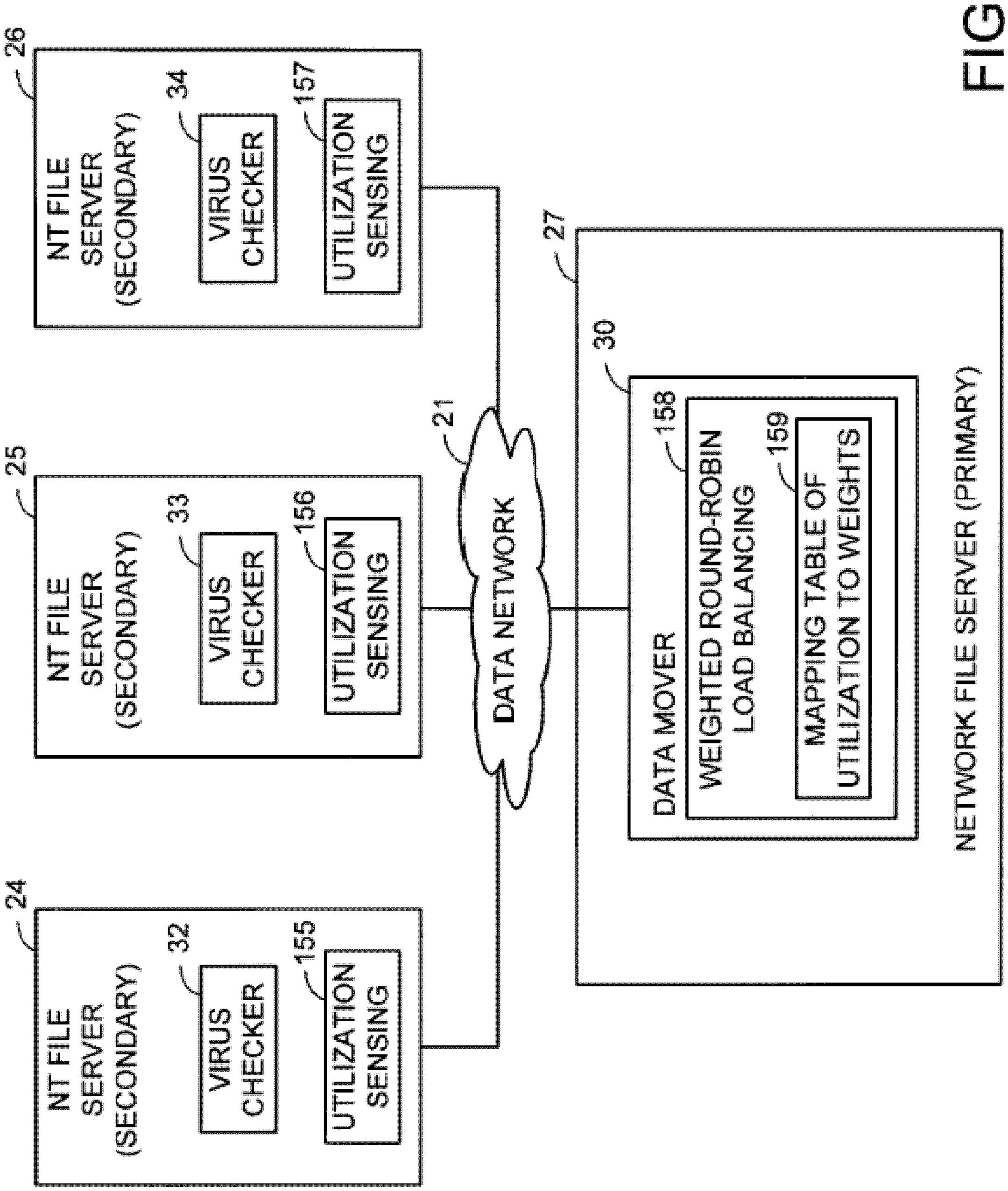
NT FILE SERVER (SECONDARY)
VIRUS CHECKER
UTILIZATION SENSING
24
32
155
25
33
156
26
34
157
21
DATA NETWORK
27
30
DATA MOVER
158
WEIGHTED ROUND-ROBIN LOAD BALANCING
159
MAPPING TABLE OF UTILIZATION TO WEIGHTS
NETWORK FILE SERVER (PRIMARY)

FIG. 14

program 158 uses the weights W[i] for distributing the virus scan requests to the virus checkers 32, 33, 34 in the NT file servers 24, 25, and 26. (Specification, page 35, lines 11-13.)

For example, appellant's FIG. 15, as reproduced below, shows the load balancing procedure used in the system of FIG. 14. (Specification, page 35, lines 14-15.) In a first step 161 of FIG. 15, each virus checker (i) computes a utilization α[i] indicating the loading on the virus checker (i) as a percentage of the saturation level of the virus checker. (Specification, page 35, lines 15-17.) Next, in step 162, the data mover collects the utilization α[i] from each of the virus checkers. (Specification, page 35, lines 17-18.) In step 163, the data mover applies a mapping function to convert each utilization α[i] to a respective weight W[i] estimated to cause a balancing of the loading upon the virus checkers. (Specification, page 36, lines 1-3.) In step 164, the data mover uses the weights W[i] for weighted round-robin load balancing of the anti-virus scan requests from the data mover to the virus checkers. (Specification, page 36 lines 5-7.) The weights W[i] are used for load balancing in step 164 until a next heartbeat interval occurs. (Specification, page 36, lines 9-10.) In step 165, when the next heartbeat interval occurs, execution loops back to step 161. (Specification, page 36, lines 10-11.)

The invention of appellant's independent claim 11 is a method in a data processing network (FIG. 14) including distributed processing units (24, 25, and 26 in FIG. 14; specification, page 35, lines 1-7). (Specification, page 3, lines 13-14.) The method includes obtaining a respective utilization value of each distributed processing unit (step 173 in FIG. 16; specification, page 39, lines 1-2); applying a mapping function (159 in FIG. 14; specification,

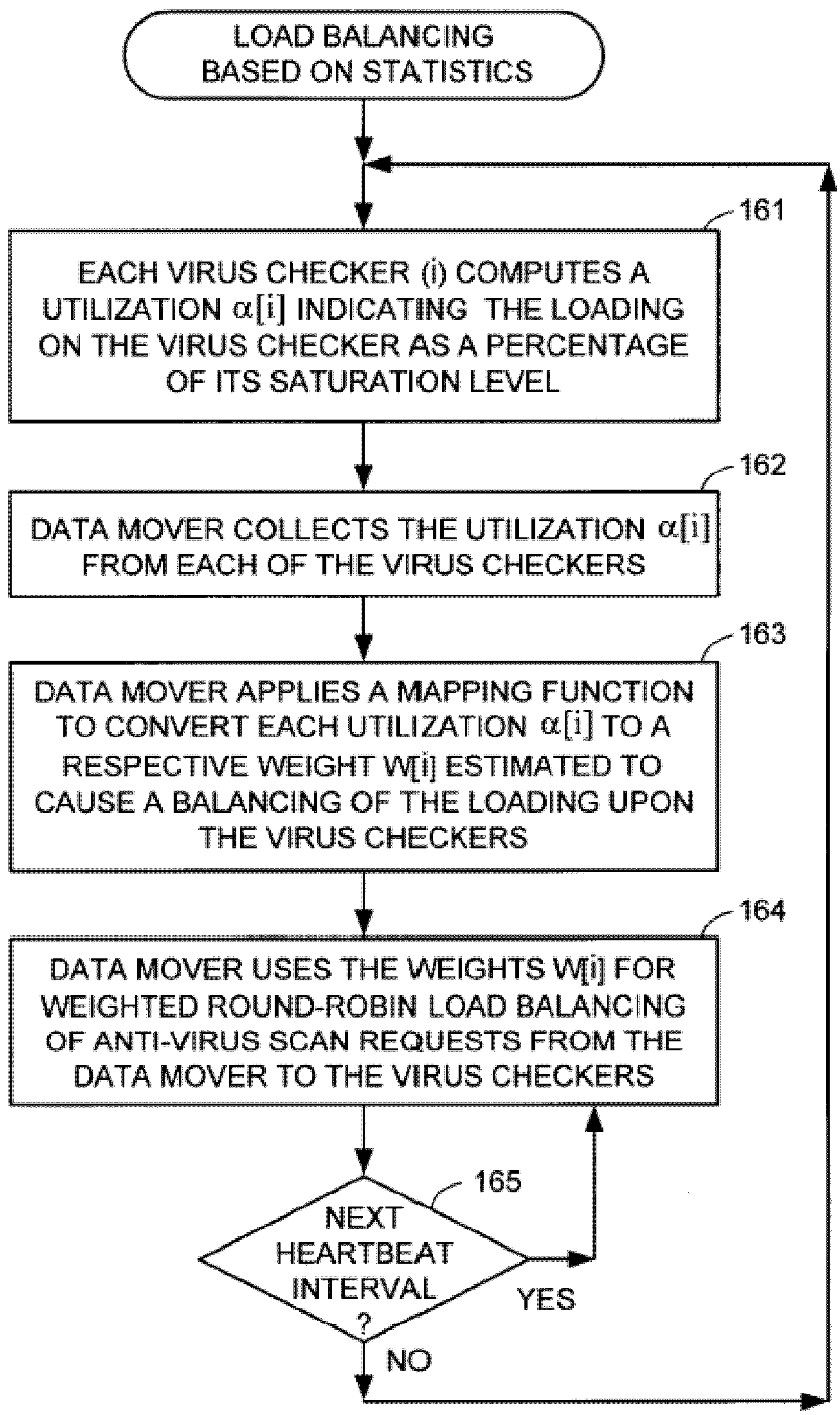
LOAD BALANCING BASED ON STATISTICS
161
EACH VIRUS CHECKER (i) COMPUTES A UTILIZATION α[i] INDICATING THE LOADING ON THE VIRUS CHECKER AS A PERCENTAGE OF ITS SATURATION LEVEL
162
DATA MOVER COLLECTS THE UTILIZATION α[i] FROM EACH OF THE VIRUS CHECKERS
163
DATA MOVER APPLIES A MAPPING FUNCTION TO CONVERT EACH UTILIZATION α[i] TO A RESPECTIVE WEIGHT W[i] ESTIMATED TO CAUSE A BALANCING OF THE LOADING UPON THE VIRUS CHECKERS
164
DATA MOVER USES THE WEIGHTS W[i] FOR WEIGHTED ROUND-ROBIN LOAD BALANCING OF ANTI-VIRUS SCAN REQUESTS FROM THE DATA MOVER TO THE VIRUS CHECKERS
165
NEXT HEARTBEAT INTERVAL ?
YES
NO

FIG. 15

page 35, lines 8-11; page 36, lines 1-5; page 37 lines 10-20) to the respective utilization value of said each distributed processing unit to obtain a respective weight for said each distributed processing unit (step 174 in FIG. 16; specification, page 39, lines 2-4), using the respective weights for the distributed processing units for producing a distribution list for distributing work requests to the distributed processing units for load balancing of the work requests upon the processing units (steps 179 and 180 in FIG. 17; specification, page 39, lines 15-19), and repetitively randomizing the distribution list during the distribution of the work requests to the distributed processing units (step 181 in FIG. 17; specification, page 40, lines 9-19). (Specification, page 3, lines 14-21.)

The invention of appellant's independent claim 12 is a method in a data processing network (FIG. 14) including a network file server (27 in FIG. 14; specification, page 7 line 24 to page 8 line 13) and a plurality of virus checking servers (24, 25, 26 in FIG. 14; appellant's specification, page 35, lines 1-7). (Specification, page 3 line 22 to page 4 line 1.) The method includes the network file server obtaining a respective utilization value of each virus checking server (step 162 in FIG. 15; specification, page 35, lines 17-18), the respective utilization value of said each virus checking server indicating a percentage of saturation of said each virus checking server (step 161 in FIG. 15; specification, page 35, lines 15-17); the network file server applying a mapping function (159 in FIG. 14; specification, page 35, lines 8-11; page 36, lines 1-5; page 37 lines 10-20) to the respective utilization value of said each virus checking server to obtain a respective weight for said each virus checking server (step 163 in FIG. 15; specification,

page 36, lines 1-3); and the network file server using the respective weights for the virus checking servers for weighted round-robin load balancing of virus checking requests from the network file server to the virus checking servers (step 164 in FIG. 15; specification, page 36 lines 5-7). (Specification, page 4 lines 1-8.)

The invention of appellant's independent 19 is a data processing system (FIG. 14) comprising distributed processing units (24, 25, 26 in FIG. 14) and a processor (30 in FIG. 14; specification, page 8 lines 11-13) coupled to the distributed processing units for distributing work requests to the distributed processing units. (Specification, page 4, lines 9-12.) The processor is programmed for obtaining a respective utilization value of each distributed processing unit (step 162 in FIG. 15; specification, page 35, lines 17-18), applying a mapping function (159 in FIG. 14; specification, page 35, lines 8-11; page 36, lines 1-5; page 37 lines 10-20) to the respective utilization value of said each distributed processing unit to obtain a respective weight for said each distributed processing unit (step 163 in FIG. 15; specification, page 36, lines 1-3), and using the respective weights for the distributed processing units for distributing work requests to the distributed processing units so that the respective weight for said each distributed processing unit specifies a respective frequency at which the work requests are distributed to said each distributed processing unit (step 164 in FIG. 15; specification, page 36 lines 5-7; page 38 lines 1-3). (Specification, page 4, lines 12-18.)

The invention of appellant's independent claim 29 is a data processing system (FIG. 14) including distributed processing units (24, 25, 26 in FIG. 14) and a processor (30 in FIG. 14; specification, page 8 lines 11-13) coupled to the distributed processing units for distributing work requests to the distributed processing units. (Specification, page 4, lines 18-22.) The processor is programmed for obtaining a respective utilization value of each distributed processing unit (step 173 in FIG. 16; specification, page 39, lines 1-2), applying a mapping function (159 in FIG. 14; specification, page 35, lines 8-11; page 36, lines 1-5; page 37 lines 10-20) to the respective utilization value of said each distributed processing unit to obtain a respective weight for said each distributed processing unit (step 174 in FIG. 16; specification, page 39, lines 2-4), using the respective weights for the distributed processing units for producing a distribution list for distributing work requests to the distributed processing units for load balancing of the work requests upon the processing units (steps 179 and 180 in FIG. 17; specification, page 39, lines 15-19), and repetitively randomizing the distribution list during the distribution of the work requests to the distributed processing units (step 181 in FIG. 17; specification, page 40, lines 9-19). (Specification, page 4 line 22 to page 5 line 6.)

The invention of appellant's independent claim 30 is a data processing system (FIG. 14) including virus checking servers (24, 25, 26 in FIG. 14; specification, page 35, lines 1-7) and a network file server (27 in FIG. 14; specification, page 7 line 24 to page 8 line 13) coupled to the virus checking servers for distributing virus checking requests to the virus checking servers. (Specification, page 5, lines 7-10.) The network file server is programmed for obtaining a

respective utilization value of each virus checking server (step 162 in FIG. 15; specification, page 35, lines 17-18), the respective utilization value of said each virus checking server indicating a percentage of saturation of said each virus checking server (step 161 in FIG. 15; specification, page 35, lines 15-17), applying a mapping function (159 in FIG. 14; specification, page 35, lines 8-11; page 36, lines 1-5; page 37 lines 10-20) to the respective utilization value of said each virus checking server to obtain a respective weight for said each virus checking server (step 163 in FIG. 15; specification, page 36, lines 1-3), and using the respective weights for the virus checking servers for weighted round-robin load balancing of virus checking requests from the network file server to the virus checking servers (step 164 in FIG. 15; specification, page 36 lines 5-7). (Specification, page 5 lines 7-16.)

Appellant's dependent claims 5 and 15 further specify that the respective weight for said each distributed processing unit is programmed into a mapping table (159 in FIG. 14; specification, page 35, lines 8-11; page 36, lines 1-5; page 37 lines 10-20), and the mapping function is applied to the respective utilization value of said each distributed processing unit to obtain the respective weight for said each distributed processing unit by indexing the mapping table with the respective utilization value of said each distributed processing unit to obtain the respective weight for said each distributed processing unit (step 174 in FIG. 16; specification, page 39, lines 2-4.)

Appellant's dependent claims 7 and 25 further specify that the respective weights are used for weighted round-robin load balancing of the work requests upon the distributed processing units. (Step 164 in FIG. 15; specification, page 36 lines 5-9.)

Appellant's dependent claims 9 and 27 further specify that the respective weights for the distributed processing units are used for distributing work requests to the distributed processing units by creating a distribution list containing entries indicating the distributed processing units (steps 179 and 180 in FIG. 17; specification, page 39, lines 15-19), the respective weight for said each distributed processing unit specifying the number of the entries indicating said each distributed processing unit (specification, page 39, lines 16-17), and by randomizing the distribution list (step 180 in FIG. 17; specification, page 39, lines 17-19; page 40, lines 9-12), and accessing the randomized distribution list for distributing the work requests to the distributed processing units as indicated by the entries in the randomized distribution list (step 181 in FIG. 17; specification, page 39, lines 13-15).

Appellant's dependent claims 10, 17, 28, and 35 further specify re-randomizing the distribution list for re-use once the end of the distribution list is reached during the distribution of the work requests to the distributed processing units as indicated by the entries in the randomized distribution list. (Step 181 in FIG. 17; specification, page 40, lines 9-19.)

Appellant's dependent claims 16 and 34 further specify that the respective weights for the virus checking servers are used for weighted round-robin load balancing of virus checking requests from the network file server to the virus checking servers (step 164 in FIG. 15; specification, page 36 lines 5-7) by creating a distribution list containing entries indicating the virus checking servers (steps 179 and 180 in FIG. 17; specification, page 39, lines 15-19), the respective weight for said each virus checking server specifying the number of the entries indicating said each virus checking server (specification, page 39, lines 16-17), and by randomizing the distribution list (step 180 in FIG. 17; specification, page 39, lines 17-19; page 40, lines 9-12), and accessing the randomized distribution list for distributing the virus checking requests from the network file server to the virus checking servers as indicated by the entries in the randomized distribution list (step 181 in FIG. 17; specification, page 39, lines 13-15).

Appellant's dependent claims 18 and 36 further specify that the network file server obtains the utilization values of the virus checking servers at the start of a heartbeat interval (step 173 in FIG. 16; specification, page 39 lines 1-2), randomizes the distribution list repetitively during use of the distribution list for load balancing of virus checking requests during the heartbeat interval (steps 180 and 181 in FIG. 17 specification, page 39, lines 17-19; page 40 lines 13-19), obtains new utilization values of the virus checking servers at the start of a following heartbeat interval (steps 182 of FIG. 17 looping back to step 173 of FIG. 16; specification, page 40 lines 19-20, page 39 lines 1-2), and produces a new distribution list from the new utilization values of the virus checking servers for load balancing of virus checking requests during the

following heartbeat interval (repeated step 174 of FIG. 16 and steps 179 and 180 of FIG. 17; specification, page 39, lines 2-5 and 15-19).

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

1. Whether claims 1-6 and 19-24 are unpatentable under 35 U.S.C. 103(a) over Van Rietschote et al. (U.S. Patent 7,203,944 B1) in view of Greuel et al. (U.S. Patent 7,003,564 B2), and further in view of Baratz et al. (U.S. Pat. App. Pub. 2002/0034190 A1).

2. Whether claims 7-8 and 25-26 are unpatentable under 35 U.S. 103(a) over Van Rietschote et al. (U.S. Patent 7,203,944 B1) in view of Greuel et al. (U.S. Patent 7,003,564 B2), and Baratz et al. (U.S. Pat. App. Pub. 2002/0034190 A1) and further in view of Garnett et al. (U.S. Patent 7,032,037).

3. Whether claims 9 and 27 are unpatentable under 35 U.S. 103(a) over Van Rietschote et al. (U.S. Patent 7,203,944 B1) in view of Greuel et al. (U.S. Patent 7,003,564 B2), and Baratz et al. U.S. Pat. App. Pub. 2002/0034190 A1) and further in view of Kapoor (U.S. Patent 5,884,038).

4. Whether claims 10 and 28 are unpatentable under 35 U.S. 103(a) over Van Rietschote et al. (U.S. Patent 7,203,944 B1) in view of Greuel et al. (U.S. Patent 7,003,564 B2), and Baratz et al. U.S. Pat. App. Pub. 2002/0034190 A1) and further in view of Grochowski (U.S. Patent 6,115,807).

5. Whether claims 11 and 29 are unpatentable under 35 U.S. 103(a) over Van Rietschote et al. (U.S. Patent 7,203,944 B1) in view of Greuel et al. (U.S. Patent 7,003,564 B2), and further in view of Kapoor (U.S. Patent 5,884,038).

6. Whether claims 12-15 and 30-33 are unpatentable under 35 U.S. 103(a) over Komai (U.S. Pat. App. Pub. 2003/0187711 A1) in view of Greuel et al. (U.S. Patent 7,003,564 B2), and further in view of Garnett et al. (U.S. Patent 7,032,037).

7. Whether claims 16, 18, 34 and 36 are unpatentable under 35 U.S. 103(a) over Komai (U.S. Pat. App. Pub. 2003/0187711 A1) in view of Greuel et al. (U.S. Patent 7,003,564 B2) and Garnett et al. (U.S. Patent 7,032,037), and further in view of Kapoor (U.S. Patent 5,884,038).

8. Whether claims 17 and 35 are unpatentable under 35 U.S. 103(a) over Komai (U.S. Pat. App. Pub. 2003/0187711 A1) in view of Greuel et al. (U.S. Patent 7,003,564 B2), Garnett et al. (U.S. Patent 7,032,037), and Kapoor (U.S. Patent 5,884,038), and further in view of Grochowski (U.S. Patent 6,115,807).

VII. ARGUMENT

The policy of the Patent and Trademark Office has been to follow in each and every case the standard of patentability enunciated by the Supreme Court in Graham v. John Deere Co., 148 U.S.P.Q. 459 (1966). M.P.E.P. § 2141. As stated by the Supreme Court:

> Under § 103, the scope and content of the prior art are to be determined; differences between the prior art and the claims at issue are to be ascertained; and the level of ordinary skill in the pertinent art resolved. Against this background, the obviousness or nonobviousness of the subject matter is determined. Such secondary considerations as commercial success, long felt but unsolved needs, failure of others, etc., might be utilized to give light to the circumstances surrounding the origin of the subject matter sought to be patented. As indicia of obviousness or nonobviousness, these inquiries may have relevancy.

148 U.S.P.Q. at 467.

The problem that the inventor is trying to solve must be considered in determining whether or not the invention would have been obvious. The invention as a whole embraces the structure, properties and problems it solves. In re Wright, 848 F.2d 1216, 1219, 6 U.S.P.Q.2d 1959, 1961 (Fed. Cir. 1988).

1. Claims 1-6 and 19-24 are patentable under 35 U.S.C. 103(a) over Van Rietschote et al. (U.S. Patent 7,203,944 B1) in view of Greuel et al. (U.S. Patent 7,003,564 B2), and further in view of Baratz et al. (U.S. Pat. App. Pub. 2002/0034190 A1).

Claims 1-4, 6, 19-22, 24

On page 2 of the Final Official Action, claims 1-6 and 19-24 were rejected under 35 U.S.C. 103(a) as being unpatentable over Van Rietschote et al. (U.S. Patent 7,203,944 B1) in view of Greuel et al. U.S. 7,003,564 B2), and further in view of Baratz et al. U.S. 2002/0034190 A1). Appellant respectfully traverses and respectfully submits that it would not have been obvious to one of ordinary skill in the art to modify Van Rietschote in view of Greuel and Baratz in order to reconstruct the appellant's invention of claims 1-6 and 19-24. Van Rietschote appears to entirely satisfactory for its intended purpose of balancing load caused by virtual machines upon distributed processing units (computer systems 10A, 10B, …, 10N) in a data processing network. Van Rietschote balances this loading in a substantially different way from the method of appellant's claims 1-6 and 19-24.

Van Rietschote balances the load caused by the virtual machines upon the distributed processing units by calculating the load of each virtual machine according to the resources that it uses. (Van Rietschote, col. 8, lines 35-37.) For example, the load is a weighted sum of CPU time, I/O activity, and memory consumed. (FIG. 4.) The CPU time can be a percentage of the total execution time that has been used for the virtual machine. (Col. 10 lines 58-62.) The virtual machines are migrated among the computer systems to balance load caused by the virtual machines. (Title.) The VM migration code on each computer system 10A-10N may be activated periodically, and may randomly select another computer system with which to compare loads and to potentially migrate a virtual machine. Over time, the periodic random selecting by each computer system may lead to relative balance in the loads on the computer systems. (Col. 5,

lines 30-37.) By selecting a virtual machine having a load that is close to 1/2 of the difference between the requesting computer system's load and the selected computer system's load, the VM migration code may approximate evening the load between the requesting computer system and the selected computer system.

In contrast to Van Rietschote, the method of appellant's claim 1 performs load balancing by a three-step process of obtaining respective utilization value of each distributed processing unit; applying a mapping function to the respective utilization value of said each distributed processing unit to obtain a respective weight for said each distributed processing unit; and using the respective weights for the distributed processing units for distributing work requests to the distributed processing units so that the respective weight for said each distributed processing unit specifies a respective frequency at which the work requests are distributed to said each distributed processing unit.

Regarding differences between appellant's claim 1 and Van Rietschote, paragraph 6 on page 4 of the final Official Action recognizes that "Van does not explicitly teach obtaining a respective weight by using a mapping function …" Paragraph 6 on page 5 of the final Official Action further cites Greuel for "obtaining a respective weight by using a mapping function {for each system variable, a mapping by which a raw data value associated with the corresponding system variable is mapped to a score, and for each system variable, a weight, col. 2, lines 39-42/ a score mapped from the CPU utilization, col. 5, lines 4-29.)." However, Greuel is directed to a method and apparatus for customizably calculating and displaying health of a computer network. (Gruel, Title.) The health of the computer network is indicated by a composite score computed

according to a composite score definition. The composite score definition preferably comprises, for each of N system variables, a mapping and a weight. (Greuel, Abstract.) Greuel col. 2, lines 37-42 say: "The composite score definition comprises a list of N different system variables; for each system variable, a mapping by which a raw data value associated with the corresponding system variable is mapped to a score; and for each system variable, a weight;". Greuel's apparatus for calculating the composite score includes "means for converting each raw data value associated with a corresponding system variable into a score in accordance with its associated mapping, whereby N scores result; a means for combining N scores in a weighted proportion according to their respective weights, so as to result in a composite score." (Greuel, col. 2, lines 31-36.) Thus, the overall score is computed as a weighted average of the N scores. (See, e.g., Greuel col. 5 lines 7-10.) Thus, Greuel does not obtain a respective weight by using a mapping function. Instead, Greuel uses a mapping function to obtain a score from a raw data value associated with a corresponding system variable. Greuel then computes an overall composite score as a weighted average of N such scores for N different system variables.

Paragraph 8 on page 4 of the final Official Action further recognizes that "Van and Greuel do not explicitly teach the respective weight for said each distributed processing unit specifies a respective frequency at which the work requests are distributed to said each distributed processing unit." Paragraph 8 on pages 4-5 of the final Official Action cites Baratz paragraphs 47-48 for a teaching of such weights. However, these paragraphs say:

> [0047] According to the present invention there is provided an improved communication network wherein a plurality of users communicate via a plurality of network elements connected by a plurality of signaling links, the improvement including: (a) at least one signaling mediation probe for monitoring usage of the network; and (b) an availability server for predicting, based on the monitored usage, a forthcoming time interval wherein sufficient network elements will be available for sending a message to one of the users.
>
> [0048] The present invention adds, to a network such as network 100, two new types of hardware: one or more signaling mediation probe and an availability server. The mediation probes monitor network usage, either by monitoring usage of one or more of the signal links, or by receiving usage information from one or more of the network elements, or by both monitoring usage of one or more of the signal links and receiving usage information from one or more of the network elements. Based on this monitoring, the availability server predicts a forthcoming time interval during which sufficient network elements will be available to send a predetermined message to one or more users. When that time interval arrives, the availability server initiates the sending of the message. "Initiating" the sending of the message includes at least eh following possibilities: either the availability server itself sends the message to the user or users, or the availability server triggers the sending of the message to the user or the users by a different device.

Appellant respectfully submits that there are further differences between the subject matter of appellant's claim 1 and the cited references. For example, page 3 of the final Official Action says: "Thus, Van inherently and obviously to one of ordinary skill in the art discloses a respective … utilization value of each distributed processing unit is obtaining." However, there

should not be confusion of the virtual machines of Van with the computer systems in Van if the virtual machines are to be considered work requests that are distributed to the computer systems of Van as is done in the last paragraph on page 3 of the final Official Action. If the virtual machines of Van are considered to be work requests and the computer systems of Van are considered to be distributed processing units, then the calculating of the load of each virtual machine according to the resources that it uses in Van is not properly considered to be "obtaining a utilization value of each distributed processing unit" and "applying a mapping function to the respective utilization value of each distributed processing unit to obtain a respective weight of said each distributed processing unit."

In view of the differences between appellant's claim 1 and Van, Greuel, and Baratz as set out above, the fact that Van discloses a substantially different method of load balancing and is entirely suitable for performing its intended function as set out above, and the different fields of use, objectives, and problems addressed by Van, Greuel, and Baratz, it is respectfully submitted that improper hindsight would be required to pick and choose the particular elements of appellant's claims 1-6 and 19-24 for combination and modification as required to reconstruct the invention of appellant's claims 1-6 and 19-24. Greuel et al. is directed to a method and apparatus for customizably calculating and displaying health of a computer network (Greuel, Title), and does not appear to concern load balancing. Baratz is directed to managing communication in a cellular network (such as GSM networks, GPRS networks, IS95-B networks, and HSCSD networks) to achieve better use of the network capacity of a cellular

network and to enable new services based on resource availability of the cellular network by considering the resource availability of network components in controlling applications and services that can be scheduled to times when such resources are available. (Baratz, page 2, par. [0045].)

When determining whether a claim is obvious, an examiner must make "a searching comparison of the claimed invention – including all its limitations – with the teaching of the prior art." In re Ochiai, 71 F.3d 1565, 1572 (Fed. Cir. 1995) (emphasis added). Thus, "obviousness requires a suggestion of all limitations in a claim." CFMT, Inc. v. Yieldup Intern. Corp., 349 F.3d 1333, 1342 (Fed. Cir. 2003) (citing In re Royka, 490 F.2d 981, 985 (CCPA 1974)). Moreover, as the Supreme Court recently stated, "there must be some articulated reasoning with some rational underpinning to support the legal conclusion of obviousness." KSR Int'l v. Teleflex Inc., 127 S. Ct. 1727, 1741 (2007) (quoting In re Kahn, 441 F.3d 977, 988 (Fed. Cir. 2006) (emphasis added)). A fact finder should be aware of the distortion caused by hindsight bias and must be cautious of arguments reliant upon ex post reasoning. Id., citing Graham, 383 U.S. at 36 (warning against a "temptation to read into the prior art the teachings of the invention in issue" and instructing courts to "guard against slipping into the use of hindsight.").

Claims 5 and 23

With respect to appellant's dependent claims 5 and 23, it is not seen where Greuel table 233 in FIG. 2B shows the respective weight or col. 4 or col. 5 describes that the respective weight is programmed into a mapping table. Instead, Greuel col. 7 lines 16-19 says the weights are specified

by the composite health score definition 305. In addition, the hyperlinks in the interface health column of the table 233 as mentioned in col. 5 lines 29-45 are not respective weights for distributed processing units. Moreover, there is no suggestion that the hyperlinks point to or are otherwise related to respective weights for distributed processing units.

2. Claims 7-8 and 25-26 are patentable under 35 U.S. 103(a) over Van Rietschote et al. (U.S. Patent 7,203,944 B1) in view of Greuel et al. (U.S. Patent 7,003,564 B2), and Baratz et al. (U.S. Pat. App. Pub. 2002/0034190 A1) and further in view of Garnett et al. (U.S. Patent 7,032,037).

In paragraph 14 on page 7 of the final Official Action, appellant's dependent claims 7-8 and 25-26 were rejected under 35 U.S. 103(a) as being unpatentable over Van Rietschote in view of Greuel and further in view of Baratz as applied to claims 1 and 19 above, and further in view of Garnett. Appellant respectfully traverses, and respectfully submits that Garnett does not provide the elements of appellant's base claims 1 and 19 that are missing from Van Rietschote and Greuel and Baratz as discussed above, nor does Garnett provide sufficient additional motivation for reconstructing appellant's invention of the base claims 1 and 19 from Van Rietschote and Greuel and Baratz.

3. Claims 9 and 27 are patentable under 35 U.S. 103(a) over Van Rietschote et al. (U.S. Patent 7,203,944 B1) in view of Greuel et al. (U.S. Patent 7,003,564 B2), and Baratz et al. U.S. Pat. App. Pub. 2002/0034190 A1) and further in view of Kapoor (U.S. Patent 5,884,038).

In paragraph 19 on page 8 of the final Official Action, appellant's dependent claims 9 and 27 were rejected under 35 U.S. 103(a) as being unpatentable over Van Rietschote in view of Greuel, and further in view of Baratz as applied to claim 1 and 19 above, and further in view of Kapoor. Appellant respectfully traverses, and respectfully submits that Kapoor does not provide the elements of appellant's base claims 1 and 19 that are missing from Van Rietschote and Greuel and Baratz as discussed above, nor does Kapoor provide sufficient additional motivation for reconstructing appellant's invention of the base claims 1 and 19 from Van Rietschote and Greuel and Baratz.

4. Claims 10 and 28 are patentable under 35 U.S. 103(a) over Van Rietschote et al. (U.S. Patent 7,203,944 B1) in view of Greuel et al. (U.S. Patent 7,003,564 B2), and Baratz et al. U.S. Pat. App. Pub. 2002/0034190 A1) and further in view of Grochowski (U.S. Patent 6,115,807).

In paragraph 24 on page 11 of the final Official Action, appellant's dependent claims 10 and 28 were rejected under 35 U.S. 103(a) as being unpatentable over Van Rietschote in view of Greuel, and further in view of Baratz as applied to claims 1 and 19 above, and further in view of

Grochowski. Appellant respectfully traverses, and respectfully submits that Grochowski does not provide the elements of appellant's claims 1 and 19 that are missing from Van Rietschote and Greuel and Baratz as discussed above, nor does Grochowski provide sufficient additional motivation for reconstructing appellant's invention of claims 1 and 19 from Van Rietschote and Greuel and Baratz. In addition, Grochowski is directed to a static instruction decoder utilizing a circular queue to decode instructions and select instructions to be issued. (Grochowski, Title.) It is respectfully submitted that without the benefit of improper hindsight a person of ordinary skill in the data processing network art would not be looking to the superscalar processor art such as Grochowski to improve the load balancing in the data processing network of Van. Nor is it seen where Grochowski discloses the "randomizing" step or function expressly recited in appellant's dependent claims 10 and 28.

Moreover, appellant objects to the reference to Hossack U.S. 6,819,276 in paragraph 26 of the final Official Action for a special definition of rotator so that rotator means randomizer. As a matter of law, a patent applicant may be his or her own lexicographer and give special meaning to words in his or her own patent, but if the applicant's specification does not reveal any special definition for a term in an applicant's claim, the then the term must be construed according to its ordinary meaning (i.e., not a special meaning used in a patent by someone else) that the term would have to a person of ordinary skill in the art in question at the time of the invention. Phillips v. AWH Corp., 415 F.3d 1303, 1312-13 (Fed. Cir. 2005)(en banc). In addition, Hossack U.S. 6,819,276 is directed to non-analogous art of a noise shaper system of an analog-to-digital converter. Non-analogous art cannot properly be pertinent prior art under 35

U.S.C. §103. In re Clay, 966 F.2d 656, 659, 23 U.S.P.Q.2d 1058, 1061 (Fed. Cir. 1992); In re Pagliaro, 210 U.S.P.Q. 888, 892 (C.C.P.A. 1981). It should also be clear to a person of ordinary skill in the art that the rotator or barrel shifter 130 in FIG. 1 of Hossack U.S. 6,819,276 may function as a randomizer only when the barrel shifter is combined with a pseudo-random number generator 160. (See Hossack U.S. 6,819,276 col. 2 lines 18-34.) Note that anticipation is established only when a single prior art reference discloses, expressly or under the principles of inherency, each and every element of a claimed invention as well as disclosing structure which is capable of performing the recited functional limitations. RCA Corp. v. Appl. Dig Data Sys.. Inc., 730 E. 2d 1440, 1444 (Fed. Cir. 1984); WL. Gore & Assocs., Inc. v. Garlock, Inc., 721 F.2d 1540, 1554 (Fed Cir. 1983).

5. Claims 11 and 29 are patentable under 35 U.S. 103(a) over Van Rietschote et al. (U.S. Patent 7,203,944 B1) in view of Greuel et al. (U.S. Patent 7,003,564 B2), and further in view of Kapoor (U.S. Patent 5,884,038).

In paragraph 28 on page 12 of the final Official Action, appellant's independent claims 11 and 29 were rejected under 35 U.S. 103(a) as being unpatentable over Van Rietschote in view of Greuel, and further in view of Kapoor. Appellant respectfully traverses. Appellant respectfully submits that appellant's claims 11 and 29 are distinguished from the proposed combination of Van Rietschote in view of Greuel for the reasons discussed above with reference to appellant's claim 1. In short, Van Rietschote balances his loading in a substantially different

way from the method of appellant's claims 11 and 29. Neither Van Rietschote nor Greuel teaches obtaining a respective weight by using a mapping function. Van's Rietschote virtual machines should not be confused with Van's computer systems with respect to the obtaining a respective utilization value of each distributed processing unit, and distributing the work requests to the distributed processing units for load balancing of the work requests upon the processing units. Appellant also respectfully submits that Kapoor does not provide these elements of claims 11 and 19 that missing from Van Rietschote and Greuel. Nor does Kapoor provide sufficient additional motivation for reconstructing the invention of appellant's claims 11 and 29 from Van Rietschote and Greuel.

6. Claims 12-15 and 30-33 are patentable under 35 U.S. 103(a) over Komai (U.S. Pat. App. Pub. 2003/0187711 A1) in view of Greuel et al. (U.S. Patent 7,003,564 B2), and further in view of Garnett et al. (U.S. Patent 7,032,037).

In paragraph 34 on pages 15-16 of the final Official Action, appellant's claims 12-15 and 30-33 were rejected 35 U.S. 103(a) as being unpatentable over Komai in view of Greuel, and further in view of Garnett. Appellant respectfully traverses. Appellant respectfully submits that the invention of appellant's claims 12-15 and 30-33 would not have been obvious from Komai, Greuel, and Garnett for reasons given above with respect to appellant's claim 1. In short, Van Rietschote balances his loading in a substantially different way from the method of appellant's claims 12-15 and 30-33. Neither Van Rietschote nor Greuel teaches obtaining a respective

weight by using a mapping function. Van's Rietschote virtual machines should not be confused with Van's computer systems with respect to the obtaining a respective utilization value of each virus checking server, and load balancing of the virus checking requests upon the virus checking servers.

In addition, Komai discloses in FIG. 1 a virus checking program 4 for a personal computer 1, and a server 2 for schedule management to notify the virus checking program of unoccupied time in the schedule (command of startup). Komai appears to be entirely satisfactory for its intended purpose of a personal computer performing virus checking for the personal computer during unoccupied time. (See, e.g., page 1 paragraph [0007].) Thus, there is no reason to modify Komai for load balancing of the virus checking requests from a network file server to virus checking servers, for example, by a round-robin technique.

7. Claims 16, 18, 34 and 36 are patentable under 35 U.S. 103(a) over Komai (U.S. Pat. App. Pub. 2003/0187711 A1) in view of Greuel et al. (U.S. Patent 7,003,564 B2) and Garnett et al. (U.S. Patent 7,032,037), and further in view of Kapoor (U.S. Patent 5,884,038).

In paragraph 43 on page 19 of the final Official Action, appellant's claims 16, 18, 34 and 36 were rejected 35 U.S. 103(a) as being unpatentable over Komai in view of Greuel, and further in view of Garnett as applied to claims 12 and 30 above, and further in view of Kapoor. Appellant respectfully traverses. Kapoor does not provide the claim elements that are missing from the base claims 12 and 30 as discussed above. Nor does Kapoor provide sufficient

motivation for combining and modifying Komai, Greuel, and Garnet to reconstruct the invention of the base claims 12 and 30.

8. Claims 17 and 35 are patentable under 35 U.S. 103(a) over Komai (U.S. Pat. App. Pub. 2003/0187711 A1) in view of Greuel et al. (U.S. Patent 7,003,564 B2), Garnett et al. (U.S. Patent 7,032,037), and Kapoor (U.S. Patent 5,884,038), and further in view of Grochowski (U.S. Patent 6,115,807).

In paragraph 50 on page 22 of the final Official Action, appellant's claims 17 and 35 were rejected 35 U.S. 103(a) as being unpatentable over Komai (U.S. 2003/0187711 A1) in view of Greuel et al. U.S. 7,003,564 B2), and further in view of Garnett et al. (U.S. 7,032,037) as applied to claims 16 and 34 above, and further in view of Kapoor (U.S. 5,884,038, and further in view of Grochowski (U.S. 6,115,807). Appellant respectfully traverses. Neither Garnett, Kapoor, nor Grochowski provides the claim elements that are missing from the base claims 12 and 30 as discussed above. Neither Garnett, Kapoor, nor Grochowski provide sufficient motivation for combining and modifying Komai, Greuel, and Garnett to reconstruct the invention of appellant's base claims 12 and 30.

In view of the above, the rejection of the claims should be reversed.

Respectfully submitted,

/ Richard C. Auchterlonie /

Richard C. Auchterlonie
Reg. No. 30,607
NOVAK DRUCE & QUIGG, LLP
1000 Louisiana, 53rd Floor
Houston, TX 77002
Telephone 713-571-3400
Telefax 713-456-2836
Richard.Auchterlonie@novakdruce.com

VIII. CLAIMS APPENDIX

The claims involved in this appeal are as follows:

1. In a data processing network including distributed processing units, a method comprising:

obtaining a respective utilization value of each distributed processing unit;

applying a mapping function to the respective utilization value of said each distributed processing unit to obtain a respective weight for said each distributed processing unit; and

using the respective weights for the distributed processing units for distributing work requests to the distributed processing units so that the respective weight for said each distributed processing unit specifies a respective frequency at which the work requests are distributed to said each distributed processing unit.

2. The method as claimed in claim 1, wherein the respective utilization value of said each distributed processing unit is a percentage of saturation of said each distributed processing unit.

3. The method as claimed in claim 1, wherein said each distributed processing unit collects statistics for calculation of the respective utilization value of said each distributed processing unit.

4. The method as claimed in claim 1, wherein statistics for calculation of the respective utilization value of said each distributed processing unit are collected from said each distributed processing unit.

5. The method as claimed in claim 1, wherein the respective weight for said each distributed processing unit is programmed into a mapping table, and the mapping function is applied to the respective utilization value of said each distributed processing unit to obtain the respective weight for said each distributed processing unit by indexing the mapping table with the respective utilization value of said each distributed processing unit to obtain the respective weight for said each distributed processing unit.

6. The method as claimed in claim 1, wherein the mapping function is selected to provide weights estimated to cause a balancing of loading upon the distributed processing units.

7. The method as claimed in claim 1, wherein the respective weights are used for weighted round-robin load balancing of the work requests upon the distributed processing units.

8. The method as claimed in claim 7, wherein the weighted round-robin load balancing performs round-robin load balancing when the weights are equal.

9. The method as claimed in claim 1, wherein the respective weights for the distributed processing units are used for distributing work requests to the distributed processing units by creating a distribution list containing entries indicating the distributed processing units, the respective weight for said each distributed processing unit specifying the number of the entries indicating said each distributed processing unit, and by randomizing the distribution list, and accessing the randomized distribution list for distributing the work requests to the distributed processing units as indicated by the entries in the randomized distribution list.

10. The method as claimed in claim 9, which includes re-randomizing the distribution list for re-use once the end of the distribution list is reached during the distribution of the work requests to the distributed processing units as indicated by the entries in the randomized distribution list.

11. In a data processing network including distributed processing units, a method comprising:

obtaining a respective utilization value of each distributed processing unit;

applying a mapping function to the respective utilization value of said each distributed processing unit to obtain a respective weight for said each distributed processing unit;

using the respective weights for the distributed processing units for producing a distribution list for distributing work requests to the distributed processing units for load balancing of the work requests upon the processing units, and

repetitively randomizing the distribution list during the distribution of the work requests to the distributed processing units.

12. In a data processing network including a network file server and a plurality of virus checking servers, a method comprising:

the network file server obtaining a respective utilization value of each virus checking server, the respective utilization value of said each virus checking server indicating a percentage of saturation of said each virus checking server;

the network file server applying a mapping function to the respective utilization value of said each virus checking server to obtain a respective weight for said each virus checking server; and

the network file server using the respective weights for the virus checking servers for weighted round-robin load balancing of virus checking requests from the network file server to the virus checking servers.

13. The method as claimed in claim 12, wherein said each virus checking server collects statistics for calculation of the respective utilization value of said each virus checking server.

14. The method as claimed in claim 12, wherein the respective weight for said each virus checking server is programmed into a mapping table, and the network file server indexes

the mapping table with said each respective utilization value to obtain the respective weight for said each virus checking server.

15. The method as claimed in claim 12, wherein the weighted round-robin load balancing performs round-robin load balancing when the weights are equal.

16. The method as claimed in claim 12, wherein the respective weights for the virus checking servers are used for weighted round-robin load balancing of virus checking requests from the network file server to the virus checking servers by creating a distribution list containing entries indicating the virus checking servers, the respective weight for said each virus checking server specifying the number of the entries indicating said each virus checking server, and by randomizing the distribution list, and accessing the randomized distribution list for distributing the virus checking requests from the network file server to the virus checking servers as indicated by the entries in the randomized distribution list.

17. The method as claimed in claim 16, which includes re-randomizing the distribution list for re-use once the end of the distribution list is reached during the distributing of the work requests to the virus checking servers as indicated by the entries in the randomized distribution list.

18. The method as claimed in claim 16, wherein the network file server obtains the utilization values of the virus checking servers at the start of a heartbeat interval, randomizes the distribution list repetitively during use of the distribution list for load balancing of virus checking requests during the heartbeat interval, obtains new utilization values of the virus checking servers at the start of a following heartbeat interval, and produces a new distribution list from the new utilization values of the virus checking servers for load balancing of virus checking requests during the following heartbeat interval.

19. A data processing system comprising distributed processing units and a processor coupled to the distributed processing units for distributing work requests to the distributed processing units, the processor being programmed for:

obtaining a respective utilization value of each distributed processing unit;

applying a mapping function to the respective utilization value of said each distributed processing unit to obtain a respective weight for said each distributed processing unit; and

using the respective weights for the distributed processing units for distributing work requests to the distributed processing units so that the respective weight for said each distributed processing unit specifies a respective frequency at which the work requests are distributed to said each distributed processing unit.

20. The data processing system as claimed in claim 19, wherein the respective utilization value of said each distributed processing unit is a percentage of saturation of said each distributed processing unit.

21. The data processing system as claimed in claim 19, wherein said each distributed processing unit is programmed for collecting utilization statistics of said each distributed processing unit.

22. The data processing system as claimed in claim 19, wherein the processor is programmed for collecting utilization statistics from said each distributed processing unit.

23. The data processing system as claimed in claim 19, wherein the respective weight for said each distributed processing unit is programmed into a mapping table, and the processor is programmed to apply the mapping function to the respective utilization value of said each distributed processing unit to obtain a respective weight for said each distributed processing unit by indexing the mapping table with said each respective utilization value of said each distributed processing unit to obtain the respective weight for said each distributed processing unit.

24. The data processing system as claimed in claim 19, wherein the mapping function is programmed to produce weights estimated to cause a balancing of loading upon the distributed processing units.

25. The data processing system as claimed in claim 19, wherein the processor is programmed for using the respective weights for weighted round-robin load balancing of the work requests upon the distributed processing units.

26. The data processing system as claimed in claim 19, wherein the processor is programmed for performing round-robin load balancing of the work requests upon the distributed processing units when the weights are equal.

27. The data processing system as claimed in claim 19, wherein the processor is programmed for using the respective weights for the distributed processing units for distributing work requests to the distributed processing units by creating a distribution list containing entries indicating the distributed processing units, the respective weight for said each distributed processing unit specifying the number of the entries indicating said each distributed processing unit, and by randomizing the distribution list, and accessing the randomized distribution list for distributing the work requests to the distributed processing units as indicated by the entries in the randomized distribution list.

28. The data processing system as claimed in claim 37, wherein the processor is programmed for re-randomizing the distribution list for re-use once the end of the distribution

list is reached during the distribution of the work requests to the distributed processing units as indicated by the entries in the randomized distribution list.

29. A data processing system comprising distributed processing units and a processor coupled to the distributed processing units for distributing work requests to the distributed processing units, the processor being programmed for:

obtaining a respective utilization value of each distributed processing unit;

applying a mapping function to the respective utilization value of said each distributed processing unit to obtain a respective weight for said each distributed processing unit;

using the respective weights for the distributed processing units for producing a distribution list for distributing work requests to the distributed processing units for load balancing of the work requests upon the processing units, and

repetitively randomizing the distribution list during the distribution of the work requests to the distributed processing units.

30. A data processing system comprising virus checking servers and a network file server coupled to the virus checking servers for distributing virus checking requests to the virus checking servers, the network file server being programmed for:

obtaining a respective utilization value of each virus checking server, the respective utilization value of said each virus checking server indicating a percentage of saturation of said each virus checking server;

applying a mapping function to the respective utilization value of said each virus checking server to obtain a respective weight for said each virus checking server; and

using the respective weights for the virus checking servers for weighted round-robin load balancing of virus checking requests from the network file server to the virus checking servers.

31. The data processing system as claimed in claim 30, wherein said each virus checking server is programmed for collecting statistics for calculating the respective utilization value of said each virus checking server.

32. The data processing system as claimed in claim 30, wherein the respective weight for said each virus checking server is programmed into a mapping table, and the network file server is programmed for indexing the mapping table with said each respective utilization value of said each virus checking server to obtain the respective weight for said each virus checking server.

33. The data processing system as claimed in claim 30, wherein the network file server is programmed for performing round-robin load balancing of the virus checking requests upon the virus checking servers when the weights are equal.

34. The data processing system as claimed in claim 30, wherein the network file server is programmed for using the respective weights for the virus checking servers for

weighted round-robin load balancing of virus checking requests from the network file server to the virus checking servers by creating a distribution list containing entries indicating the virus checking servers, the respective weight for said each virus checking server specifying the number of the entries indicating said each virus checking server, and by randomizing the distribution list, and accessing the randomized distribution list for distributing the virus checking requests from the network file server to the virus checking servers as indicated by the entries in the randomized distribution list.

35. The data processing system as claimed in claim 34, wherein the network file server is programmed for re-randomizing the distribution list for re-use once the end of the distribution list is reached during the distributing of the work requests to the virus checking servers as indicated by the entries in the randomized distribution list.

36. The data processing system as claimed in claim 34, wherein the network file server is programmed for collecting utilization statistics from the virus checking servers at the start of a heartbeat interval, for randomizing the distribution list repetitively during use of the distribution list for load balancing of virus checking requests during the heartbeat interval, for collecting a new set of utilization statistics from the virus checking servers at the start of a following heartbeat interval, and for producing a new distribution list from the new set of utilization statistics for load balancing of virus checking requests during the following heartbeat interval.

IX. EVIDENCE APPENDIX

None.

X. RELATED PROCEEDINGS APPENDIX

None.